



Introduction to

ARDUINO 

How to use electronics to make your projects better!

Guest wifi:	HSNOTTS-guest	Password: hackspacebiscuits
Presentation:	http://wiki.nottinghack.co.uk/wiki/Arduino101	
Software:	http://arduino.cc/en/Main/Software	



Welcome!

- Let's get programming!
- What did we just do? What is an “Arduino” anyway?
- Serial communications (making your Arduino talk to your laptop)
- Electrical Basics: Voltage, current, resistance
- More blinking LEDs
- Using switches



Welcome!

- Lunch?
- Structure and Decisions: Order out of chaos
- Your Arduino is a wimp: giving it some muscle!
- Life isn't digital: reading real-world values
- Putting it all together



Dive in: Blink an LED

You must have this. All initialisation goes here.

You must have this. The main code goes here.

```
int ledPin = 13;           // LED connected to digital pin 13

void setup()               // run once, when the sketch starts
{
    pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

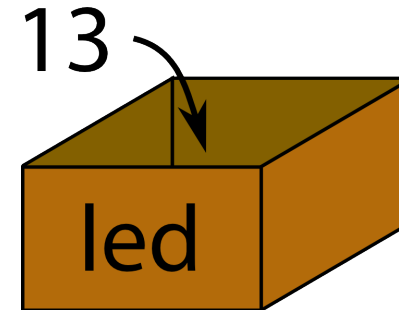
void loop()                // run over and over again
{
    digitalWrite(ledPin, HIGH); // sets the LED on
    delay(1000);                // waits for a second
    digitalWrite(ledPin, LOW);  // sets the LED off
    delay(1000);                // waits for a second
}
```



Programming Concepts

Variables : Named places (think “labelled boxes”) to store data like numbers, text, true/false.

Statement: A single line of code that does one job.



`int led = 13;`

Datatype Name Initial Value Semicolon

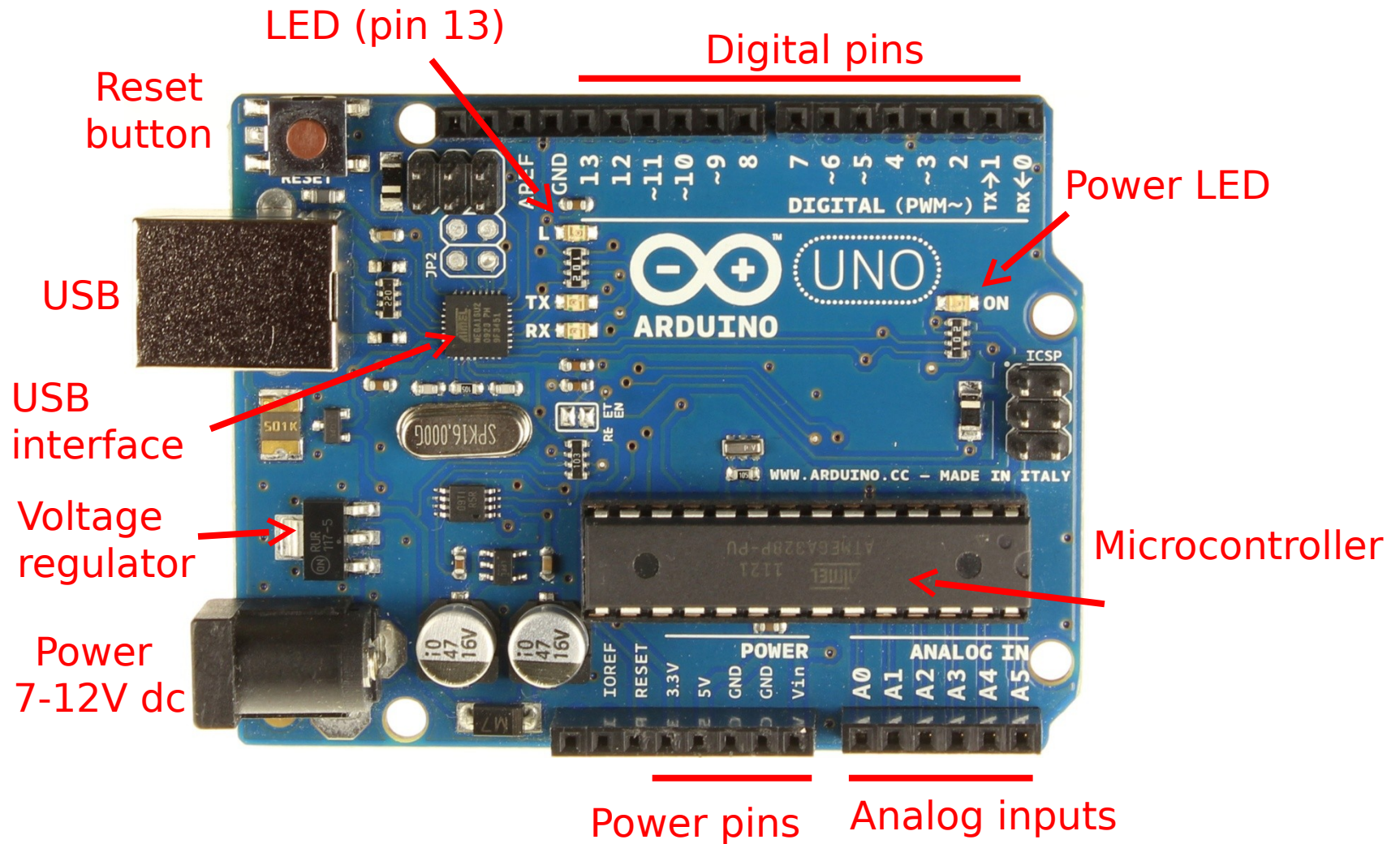
Functions: Named blocks of statements that perform a particular job

```
int getNewSensorValue(void)
{
    int value = analogRead(3);
    value = value * 4;
    return value;
}
```





The Arduino Uno





The Arduino IDE

- Arduino IDE (Integrated Development Environment)
- Program is called a “sketch”

The screenshot shows the Arduino IDE window titled "Blink | Arduino 1.0.1". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for opening files, saving, and running. The main text area displays the "Blink" sketch code, which is a standard example for turning an LED on and off. The code is as follows:

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

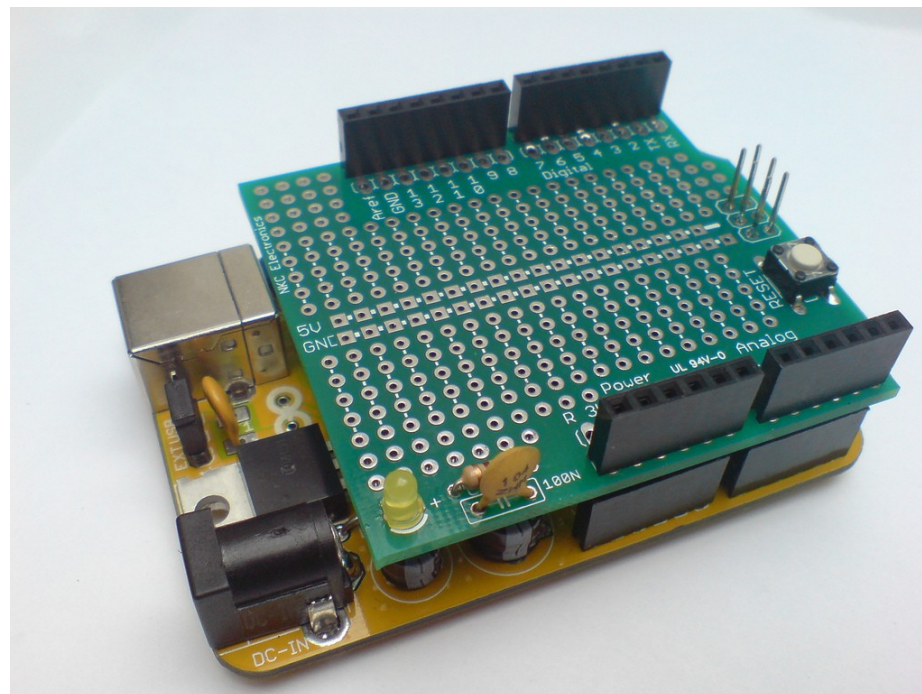
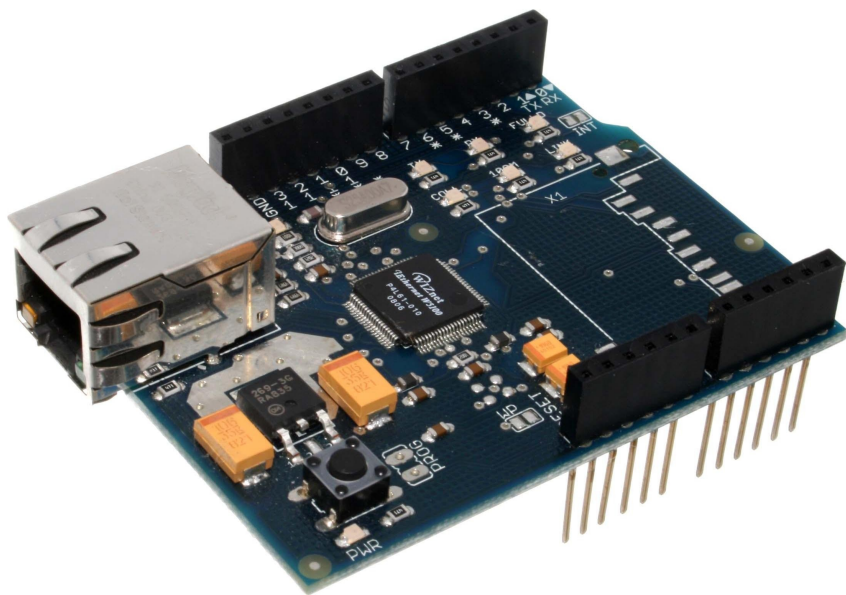
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

At the bottom of the window, a status bar indicates "1" on the left and "Arduino Nano w/ ATmega168 on COM8" on the right.



Shields Up!

Plug-in boards that expand the capabilities of your Arduino





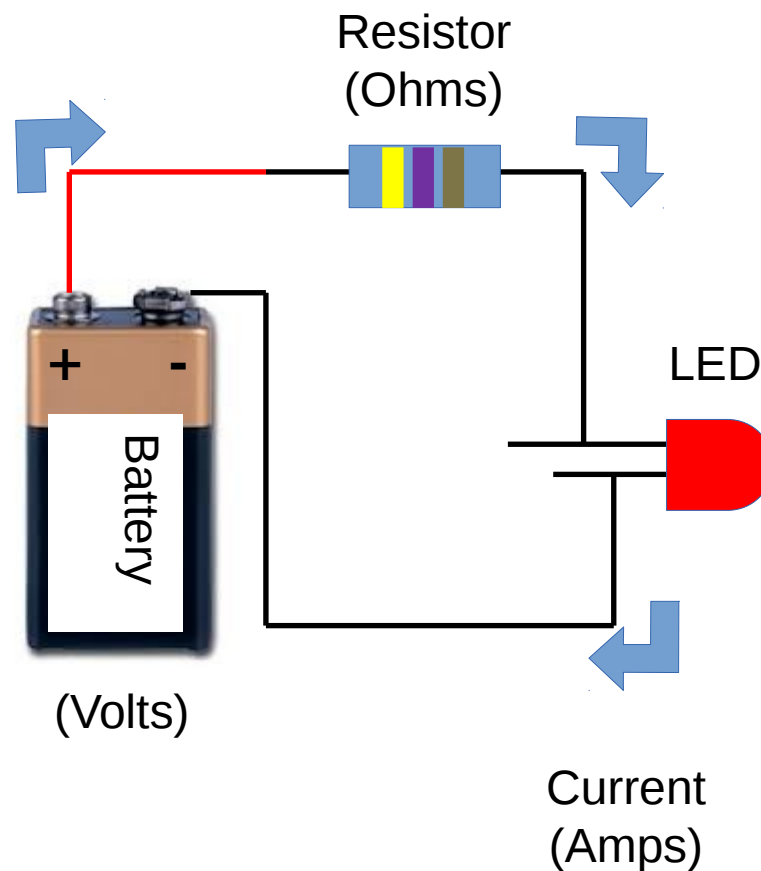
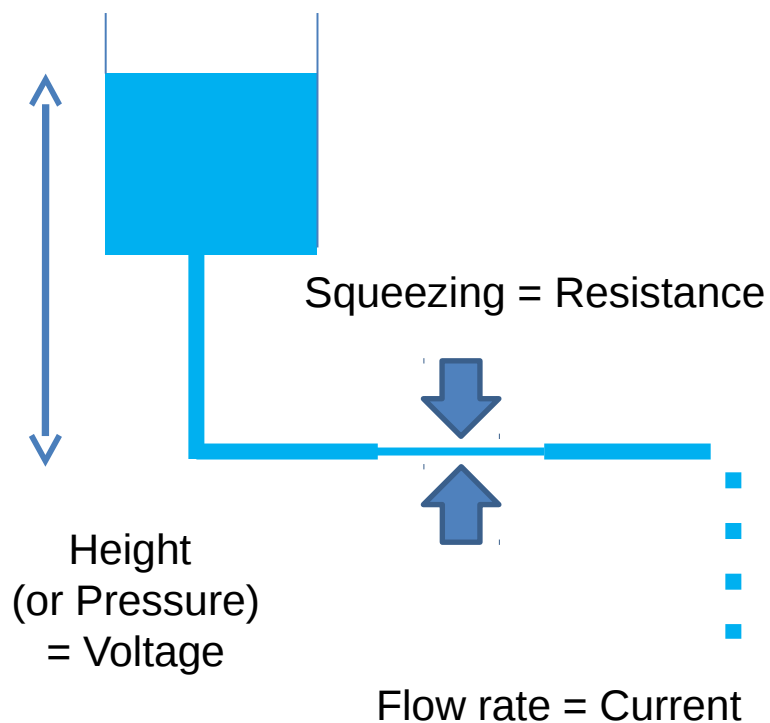
Serial Monitor

```
int ledPin = 13;           // LED connected to digital pin 13
void setup()               // run once, when the sketch starts
{
    pinMode(ledPin, OUTPUT); // sets the digital pin as output
    Serial.begin(9600);      // starts the serial port at 9600 baud
}

void loop()                // run over and over again
{
    digitalWrite(ledPin, HIGH); // sets the LED on
    Serial.println("LED on");
    delay(1000);              // waits for a second
    digitalWrite(ledPin, LOW); // sets the LED off
    Serial.println("LED off");
    delay(1000);              // waits for a second
}
```

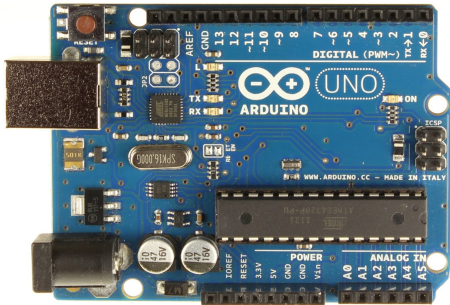


Voltage, Current, Resistance





Voltage, Current, Power



5V
0.5mA
2.5mW



360V
250A
90kW



5V
0.5A
2.5W



160V
1000A
160kW



240V
4A
960W



400kV
5000A
2000MW



Resistor Colour Codes



3600

= 3K6 +/- 5%

Multipliers:

R or Ω = 1

K = 1,000

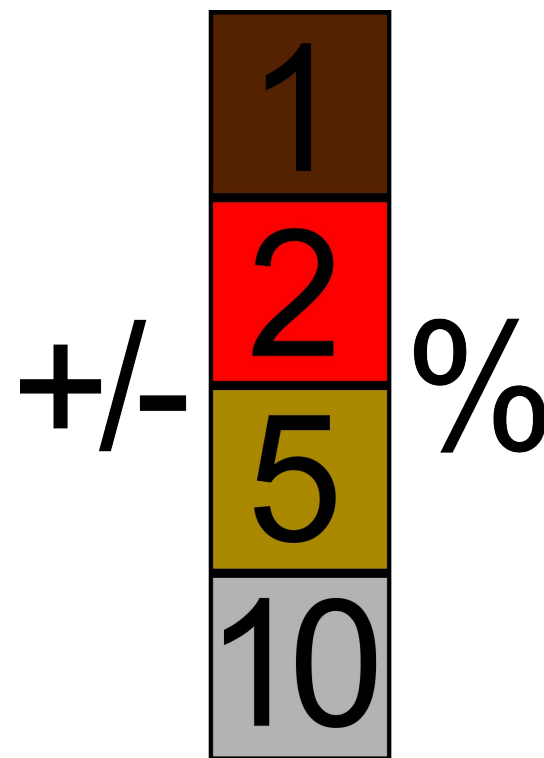
M = 1,000,000

For example:

1K = 1,000

220R = 220 Ω = 220

6M8 = 6,800,000



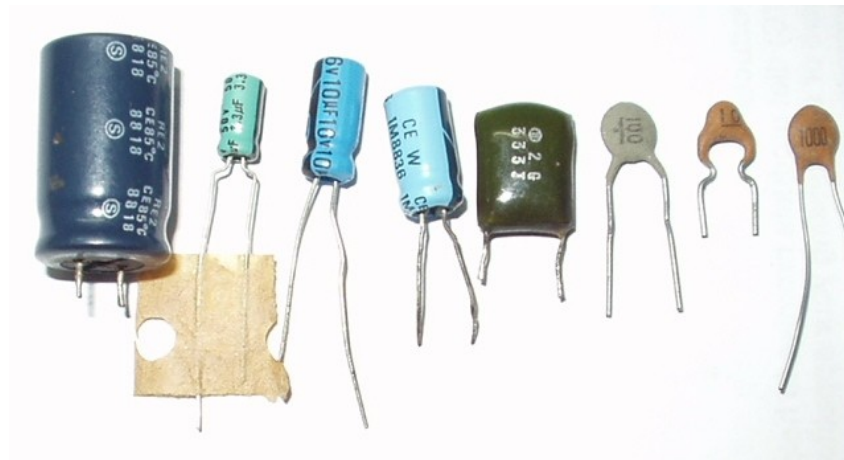


Components

Resistor



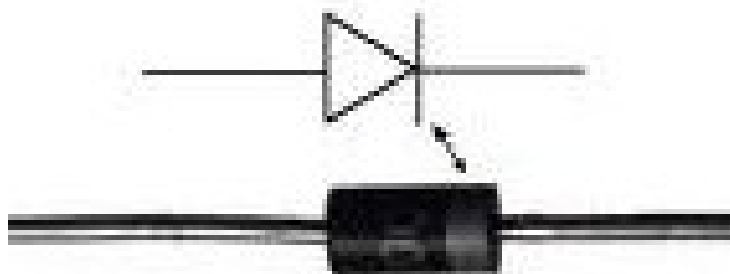
Capacitor



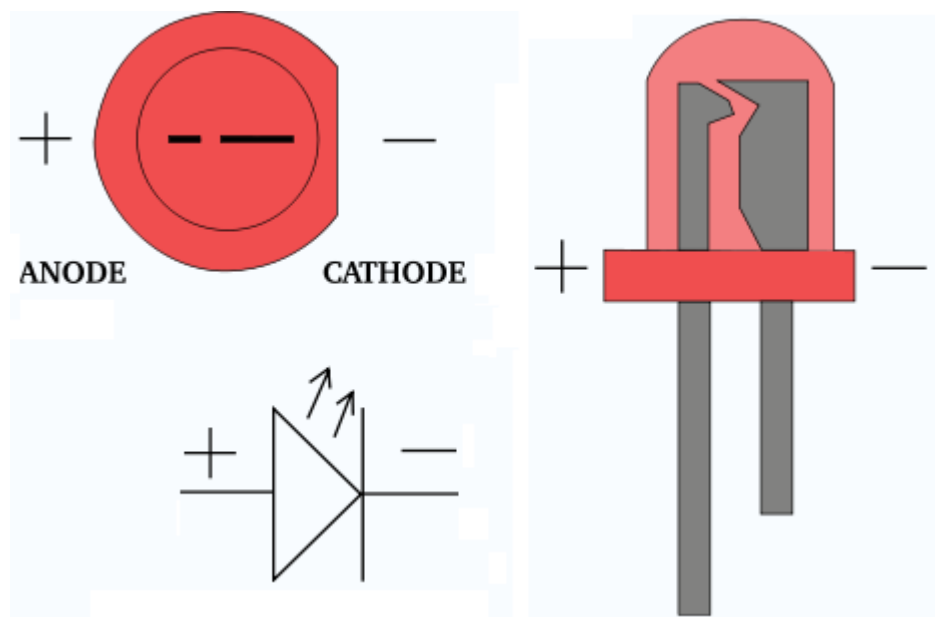


Components

Diode

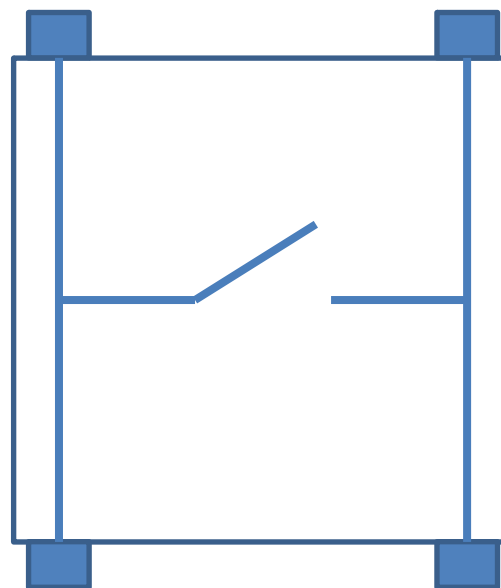
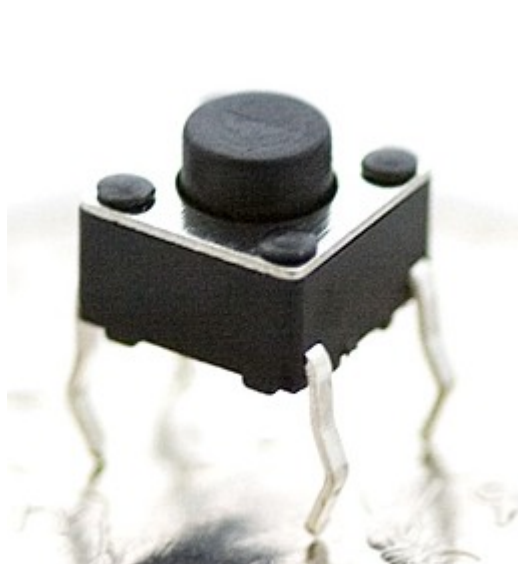


LED



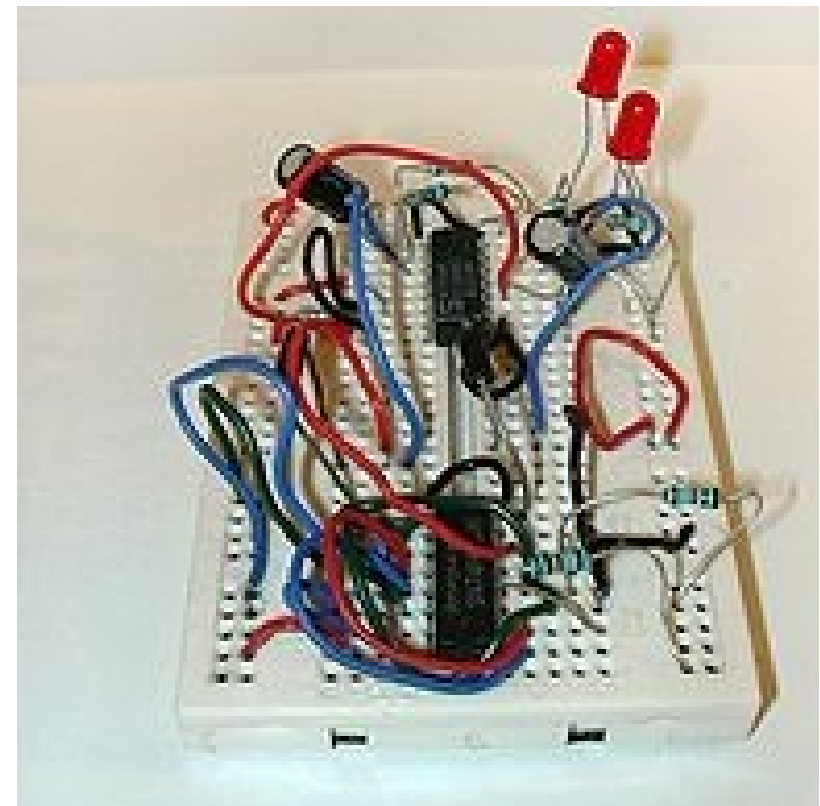
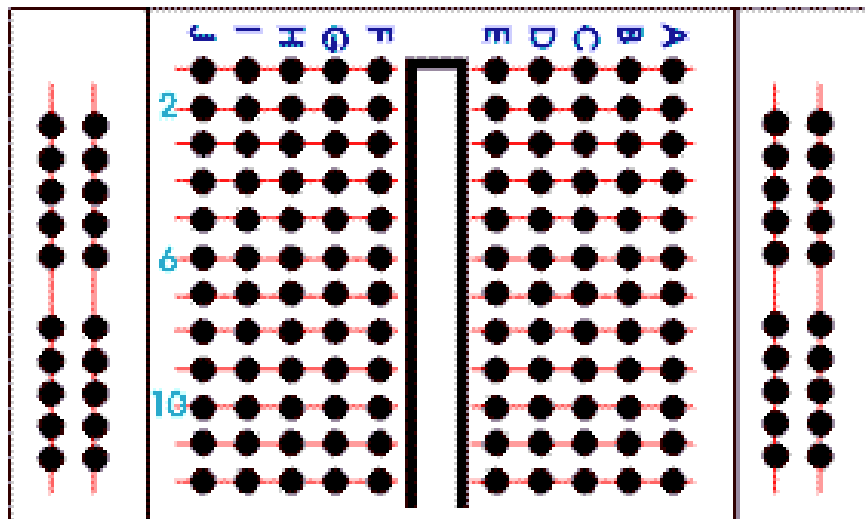


Tactile Switch





Breadboard





Another blinking LED

Alter the previous sketch:

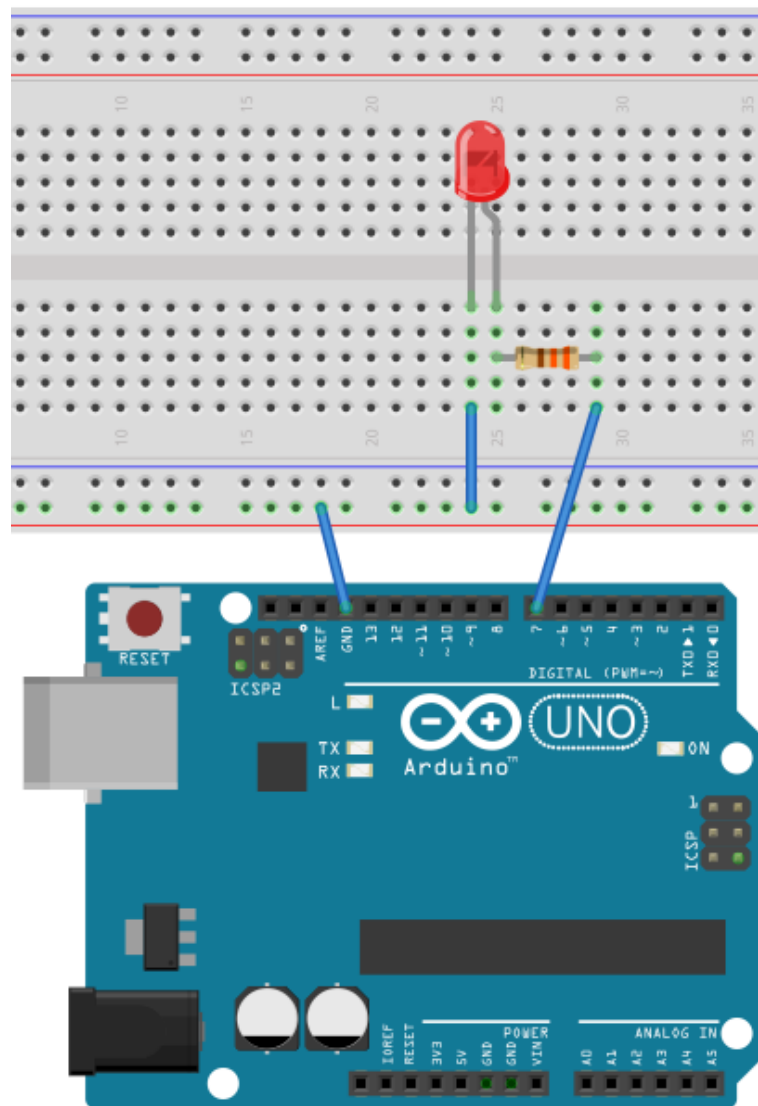
```
int ledPin = 13;
```

to

```
int ledPin = 7;
```

Exercises:

- Alter the blink frequency
- Two short blinks and one long blink
- Blink both LEDs alternately





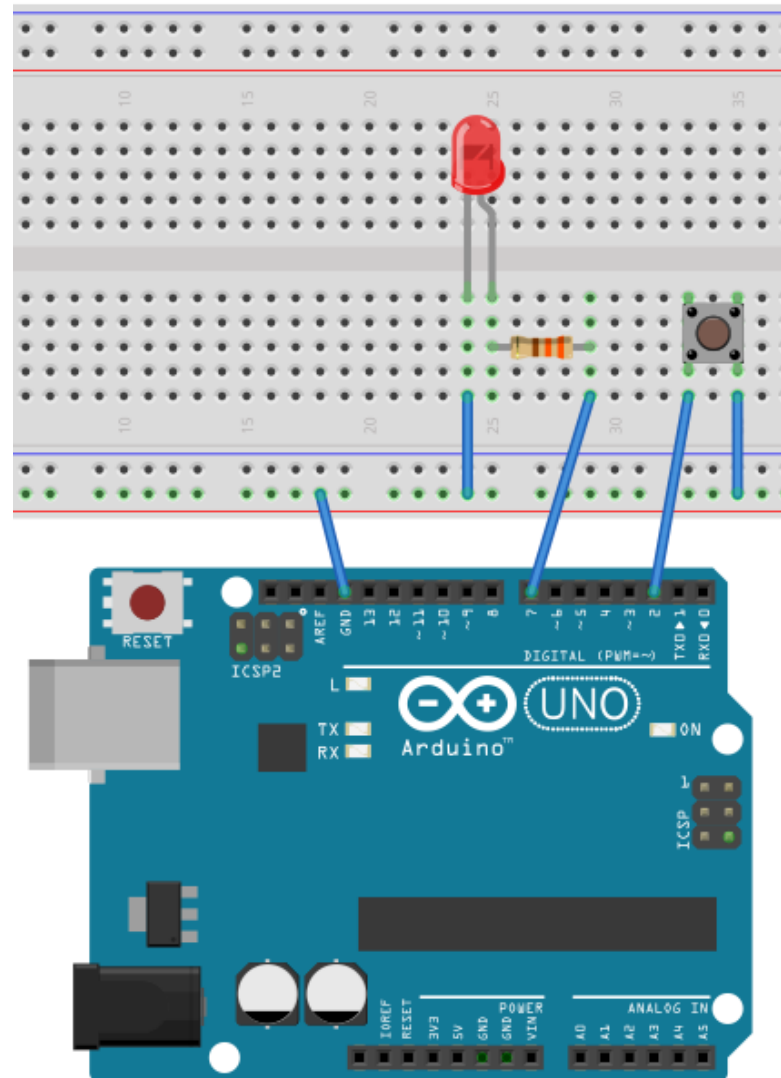
Switch controlling LED

```
int ledPin = 13;
int switchPin = 2;

// store the switch value
int switchvalue;

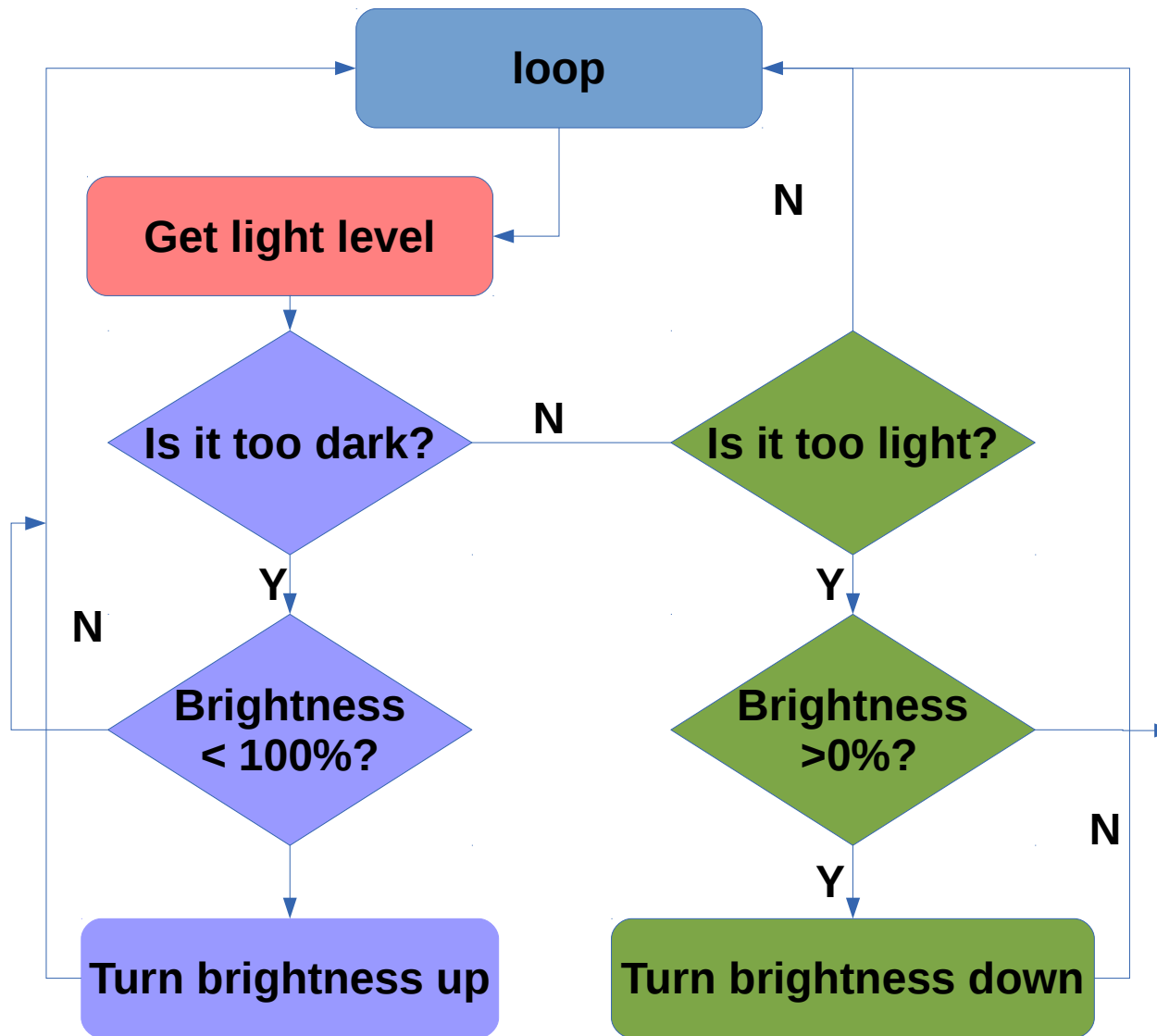
void setup ()
{
  pinMode(ledPin, OUTPUT);
  // Set the switch pin
  // as input with pull-up
  pinMode(switchPin, INPUT_PULLUP);
}

void loop ()
{
  // read the switch value
  switchvalue = digitalRead(switchPin);
  // and write it to the LED
  digitalWrite(ledPin, !switchvalue);
}
```





Making sense of your code





From Flowchart to Code

```
int lightLevel = 0;
int tooDarkLevel = 100;
int tooLightLevel = 125;

int brightness = 0;
```

```
void setup()
{
  setBrightness( brightness );
}
```

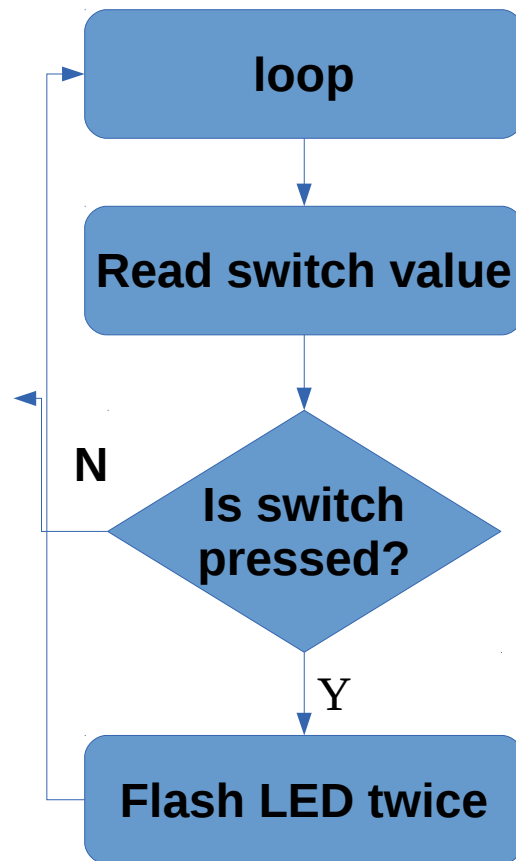
```
void loop()
{
  lightLevel = getLightValue();

  if (lightLevel < tooDarkLevel)
  {
    if (brightness < 100)
    {
      brightness = brightness + 1;
      setBrightness( brightness );
    }
  }
  else if (lightLevel > tooLightLevel)
  {
    if (brightness > 0)
    {
      brightness = brightness - 1;
      setBrightness( brightness );
    }
  }
}
```




Exercise using “if”

Blink the LED twice each time the switch is pressed.



Use “==” for comparison:
`if (switchvalue == LOW)`

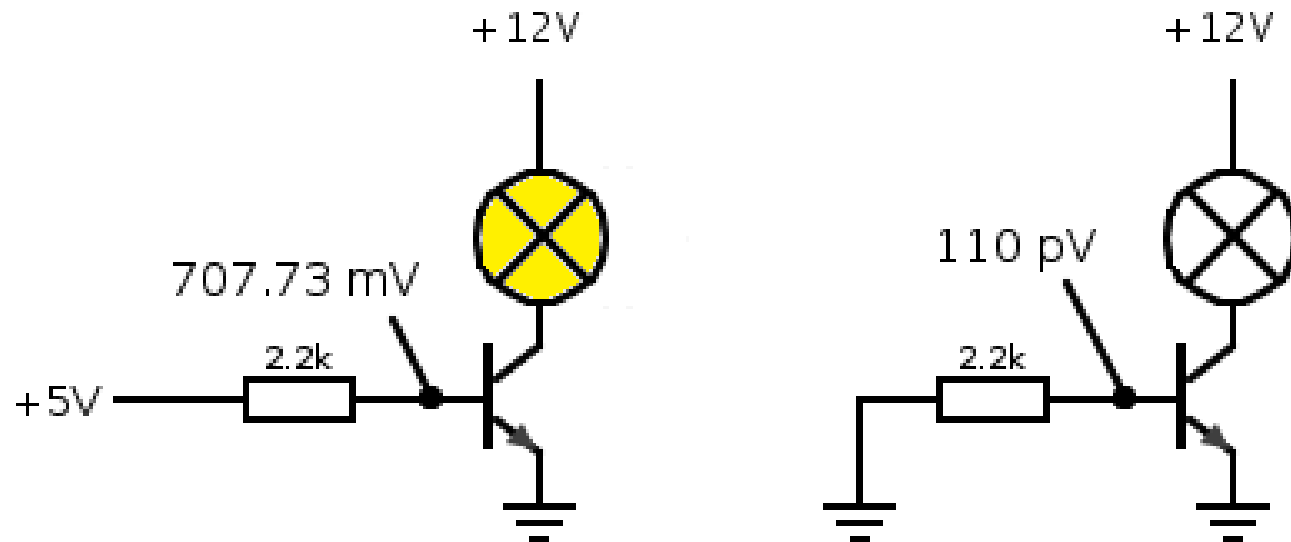
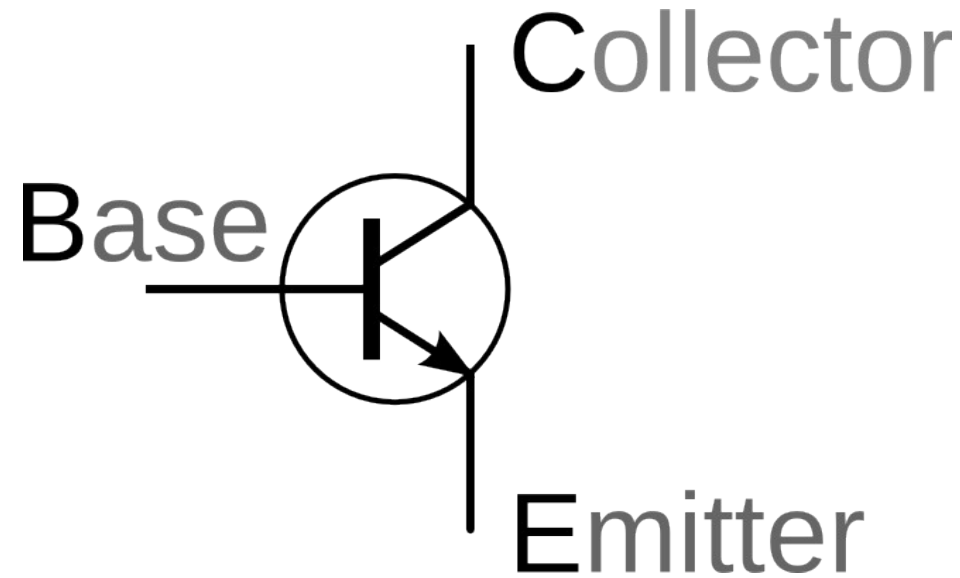


High Power Loads

- These devices need more current than the Arduino can supply.
- We use a transistor to increase the current available.
- Transistor acts as a Arduino controlled switch.
- Need protection from inductive loads (usually wound coils such as motors and relays).
- NPN Bipolar and N-channel MOSFET are commonly used with the Arduino.
- Voltage of the load (motors, etc.) can be higher than Arduino voltage.

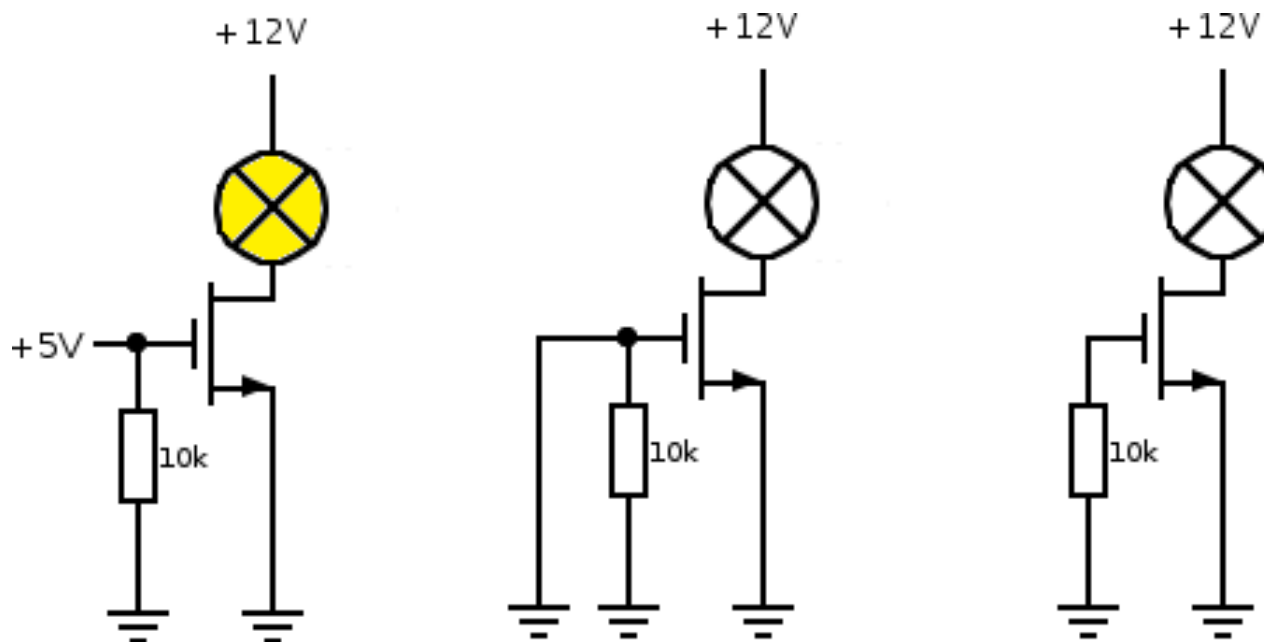
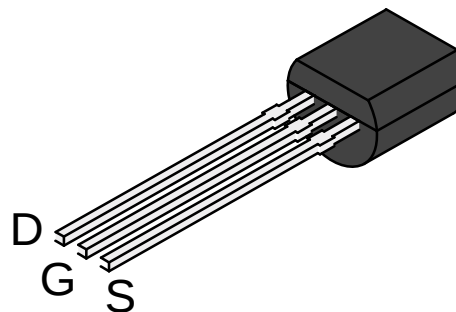
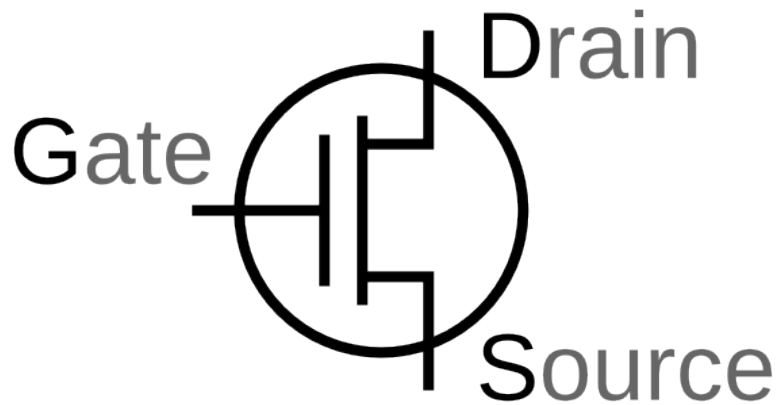


NPN Transistor





N-Channel MOSFET





Pulse Width Modulation

Drives motor at full voltage.

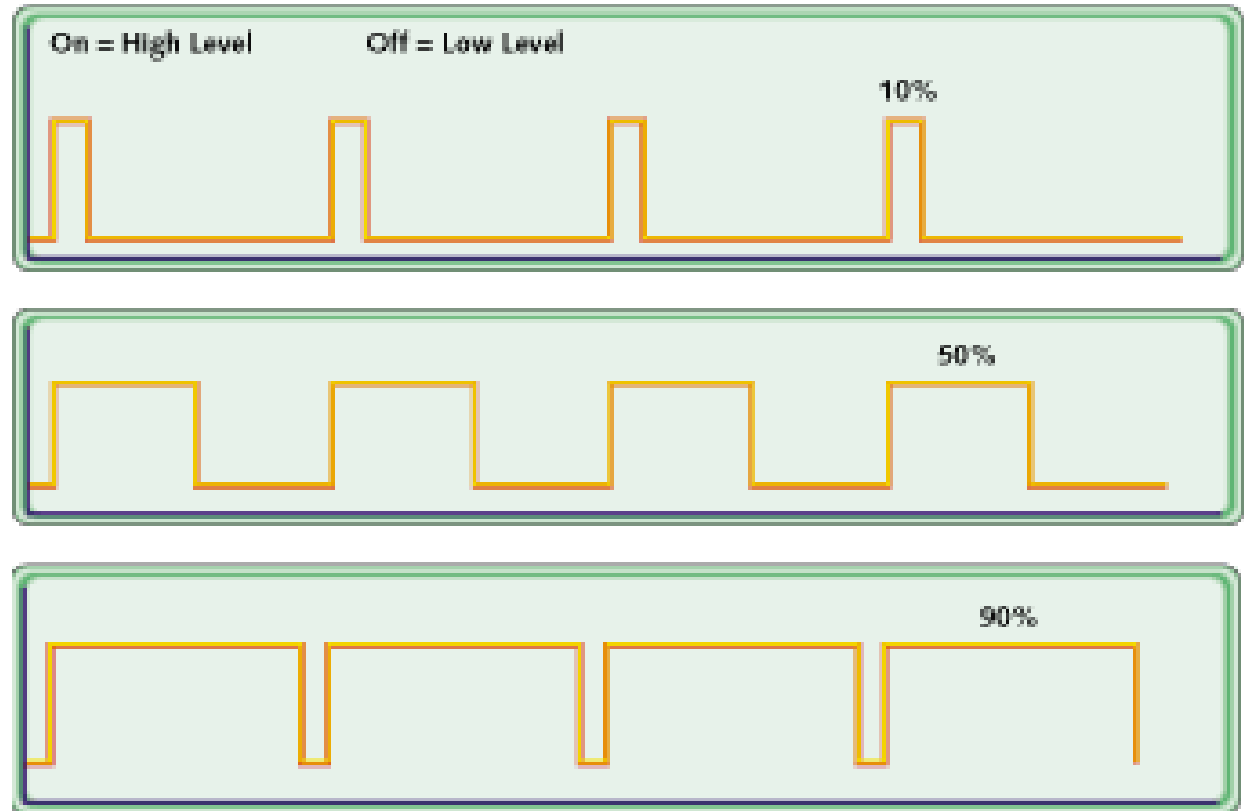
This gives high torque even at low speeds.

Used on Arduino pins 3, 5, 6, 9, 10, 11

Code:

```
analogWrite(pin,value);
```

Value 0 - 255





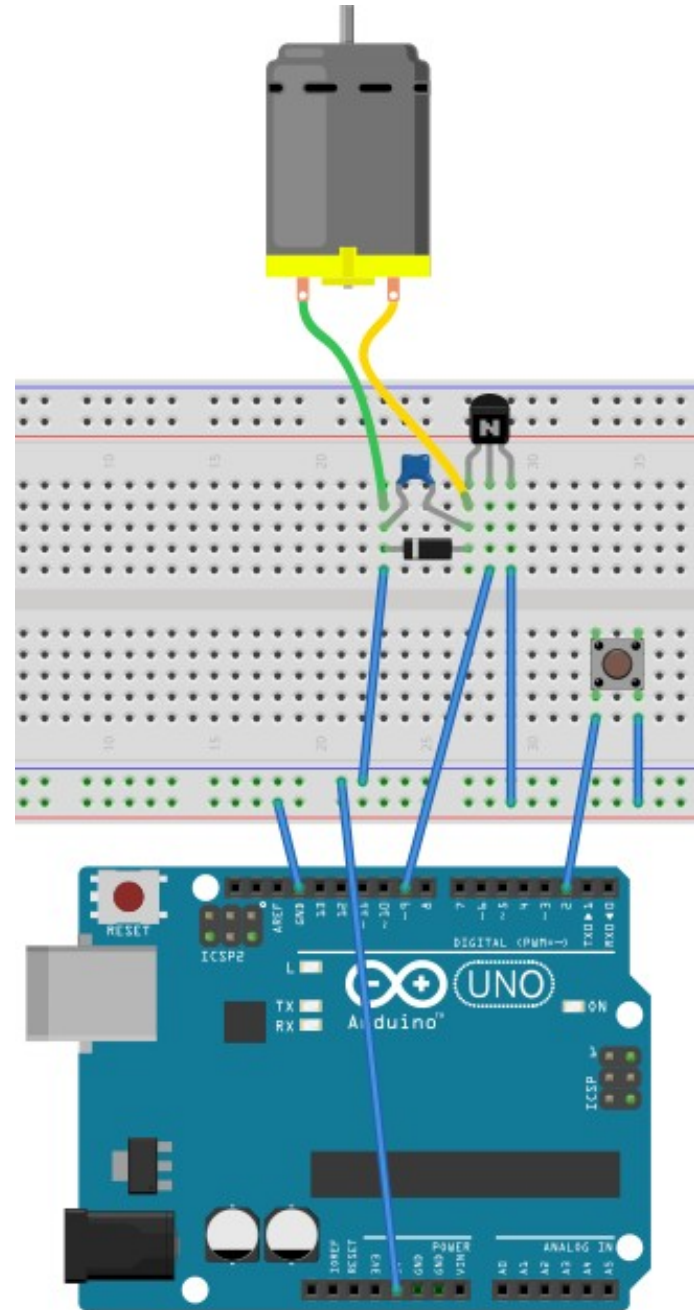
Motor Control

Motor on/off control with switch

```
void setup()
{
    pinMode(9, OUTPUT);
    pinMode(2, INPUT_PULLUP);
}

void loop()
{
    if (digitalRead(2) == 0)
    {
        analogWrite(9, 100);
    }
    else
    {
        analogWrite(9, 0);
    }
}
```

<http://wiki.nottinghack.co.uk/wiki/Arduino101>



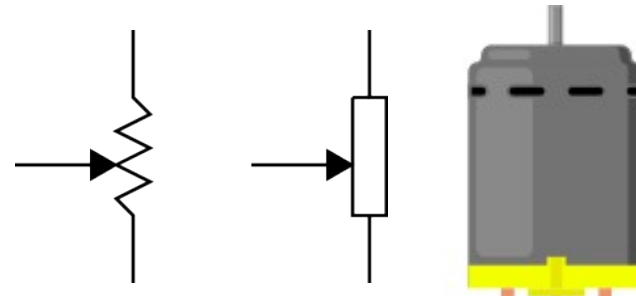


Reading real-world signals

- Arduino has 6 analog inputs
- Read with:
 - `int value = analogRead(pin);`
 - Pin is 0-5 or A0-A5
 - Value from 0 to 1023 representing voltage of 0-5V

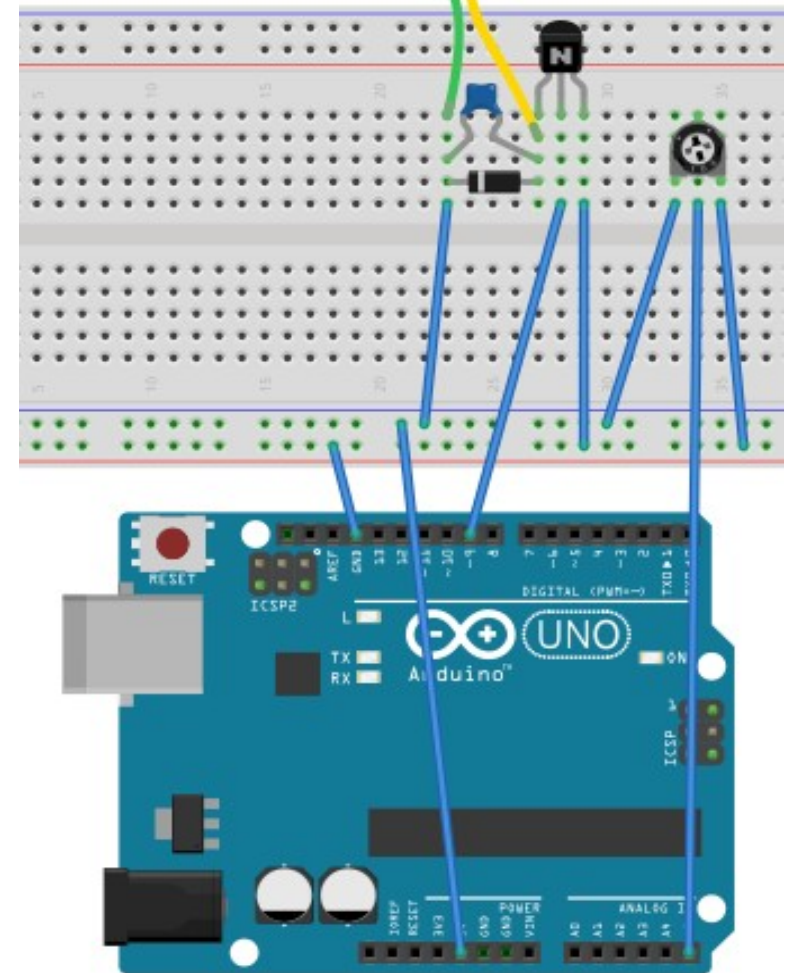


Potentiometer (Variable Resistor)



Speed control using potentiometer

```
void setup()  
{  
    pinMode(9, OUTPUT);  
}  
  
void loop()  
{  
    int light = analogRead(5);  
    analogWrite(9, light/4);  
}
```

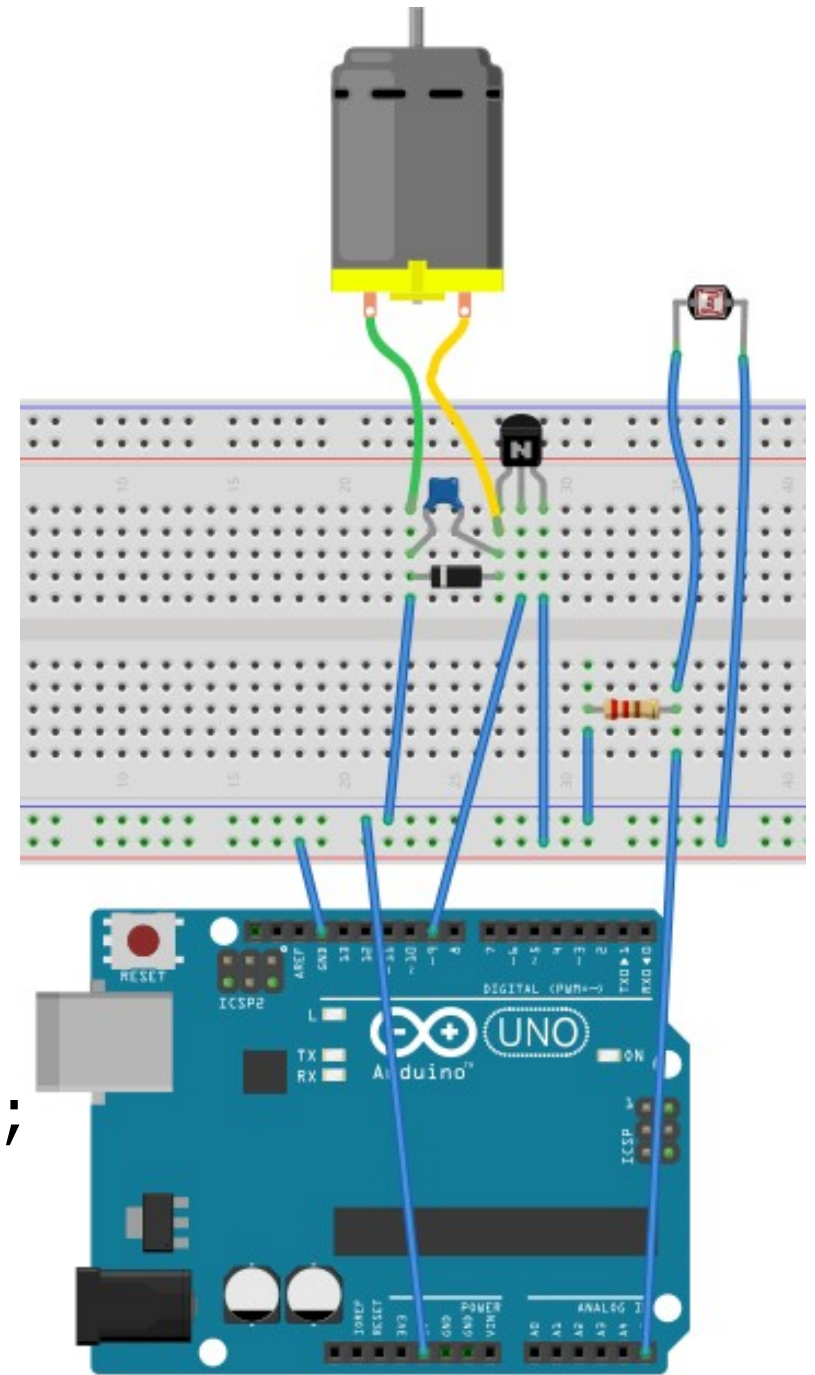




Light Dependant Resistor

Speed control according to light level

```
void setup()  
{  
    pinMode(9, OUTPUT);  
}  
  
void loop()  
{  
    int light = analogRead(5);  
    analogWrite(9, light/4);  
}
```





Challenges

- “Knightrider” LEDs
- Colour Changing RGB LED
 - Traffic lights
 - Make some noise!

Hint: use LDR and tone() function

- Line following robot
- Your own project?